

METHOD AND APPARATUS FOR MEASURING SIMILARITY  
USING MATCHING PIXEL COUNT

5 Technical Field:

The present invention relates to digital image processing and, more particularly, to a method and apparatus for measuring similarity between pixels in two different images.

10 Background Art:

15 Stereo disparity estimation is one of the most active research areas in the field of three-dimensional or stereo vision. Stereo disparity is the spatial offset or displacement between two matching pixels, where one pixel is in a reference image, the other pixel is in a search image, and both pixels correspond to the same point in physical space. The reference and search images are the images viewed by the left and right eyes, respectively.

20 In order to determine the pixel in the search image which matches a given pixel in the reference image, the coordinates of the pixels in the two images must first be established. A similarity measure is then computed between pixels in a window of predetermined size centered at the given pixel in the reference image (i.e., the reference pixel) and pixels that are candidates to match the reference pixel in a window of the same size centered at each candidate pixel in the search image. The  
25 pixel in the search image that yields the greatest value of the similarity measure is considered to be the pixel that matches the reference pixel.

30 A disparity map for the reference image is obtained by  
35 computing the similarity measure between a reference pixel and each candidate matching pixel in a search range for every pixel in the reference image. Generation of the disparity map is greatly simplified by imposition of the "epipolar constraint"

on the candidate matching pixels, whereby the candidate matching pixels are limited to those which lie on the same horizontal line as the reference pixel.

5           Figure 1 illustrates matching subject to the epipolar constraint. To determine the pixel in the search image that most closely matches the starred reference pixel, the similarity measure is computed for a window centered at the reference pixel and a window in the search image centered at  
10           a candidate matching pixel as the window in the search image is moved along a horizontal line within a search range. The candidate search pixel that yields the greatest value of the similarity measure is considered to be the pixel that matches the reference pixel. The spatial displacement between the  
15           matching pixel from the reference pixel is the stereo disparity for the reference pixel.

Similarity measures typically used to determine matching pixels include the sum of squared differences (SSD), the sum of absolute differences (SAD), and the normalized cross correlation (NCC). For each of these similarity measures, the contribution of a given search pixel to the disparity map depends on the light intensity at the given search pixel. The disparity boundary between a region within which there are large variations in light intensity therefore tends to extend  
25           into any adjacent region within which there are small variations in light intensity. This phenomenon, known as "boundary overreach", generates misleading disparity values near the boundary of an object.

30           Disclosure of Invention:

35           An object of the present invention is accordingly to suppress boundary overreach in the estimation of stereo disparity by providing a similarity measure based on a matching pixel count ("MPC"), the number of search pixels having similar intensity values, rather than on the light intensity at each

of the search pixels. Another object of the present invention is to provide an apparatus for estimating stereo disparity that efficiently implements the above method in real time.

5 One embodiment of the present invention is a method for determining the stereo disparity between a reference image and a search image for a reference pixel in the reference image, the method comprising the steps of:

10 (a) calculating a similarity measure between a reference window in the reference image, which reference window includes a set of pixels centered at the reference pixel, and each of a plurality of search windows in the search image, each of which search windows has the same shape as the reference window and is displaced from the reference window within a  
15 predetermined search range, wherein a matching pixel count, the number of pixels in the reference window at which the light intensity is similar to the light intensity at the corresponding pixels in the search window, is used as the similarity measure between the reference window and the search  
20 window; and

(b) determining a displacement between the reference window and a search window which yields the greatest value of the similarity measure as the stereo disparity for the reference pixel.

25 Another embodiment of the present invention is an apparatus for determining the stereo disparity between a reference image and a search image for a reference pixel in the reference image, the apparatus comprising:

30 (a) first means for calculating a similarity measure between a reference window in the reference image, which reference window includes a set of pixels centered at the reference pixel, and each of a plurality of search windows in the search image, each of which search windows has the same  
35 shape as the reference window and is displaced from the reference window within a predetermined search range, wherein a matching pixel count, which is the number of pixels in the

reference window which are similar in intensity to corresponding pixels in a search window, is used as the similarity measure between the reference window and the search window; and

5 (b) second means for determining a displacement between the reference window and a search window which yields a largest value of the similarity measure as the stereo disparity for the reference pixel,

10 wherein  $R(x,y)$  denotes the reference pixel, the reference window includes  $W_x \times W_y$  pixels centered at  $R(x,y)$ ,  $W_x$  and  $W_y$  being predetermined numbers, each of the search windows includes  $W_x \times W_y$  pixels centered at  $L(x+d, y)$  which is a pixel in the search image,  $d$  ranging from 0 to a predetermined number  $S_r$ , and

15 the first means includes:

(a1) a P-unit for calculating  $P(x,y,d)$ , where:

$P(x,y,d) = 1$ , if  $\text{abs}(B_R(x,y) - B_L(x+d,y)) \leq Th$  and

$P(x,y,d) = 0$ , otherwise,

20 where  $B_R(x,y)$  and  $B_L(x+d,y)$  denote the intensity values at  $R(x,y)$  and  $L(x+d,y)$ , respectively, and  $Th$  is a predetermined threshold value;

(a2) a P-buffer for storing  $P(x,y,d)$  values calculated by the P-unit; and

25 (a3) third means for determining  $MPC(x,y,d)$  values for  $d=0$  to  $S_r$  as follows:

$$MPC(x, y, d) = \sum_w P(x, y, d)$$

where  $w$  denotes the reference window and a search window; and

30 the second means includes means for selecting a  $d$  value which yields a largest  $MPC(x,y,d)$  value as the stereo disparity for  $R(x,y)$ .

Brief Description of Drawings:

Figure 1 illustrates matching subject to an epipolar constraint.

Figure 2 illustrates the comparative performance of the SAD and MPC similarity measures.

Figures 3A and 3B illustrate redundant operations involved in determining the MPC similarity measure.

Figures 4A and 4B illustrate a method of determining the MPC similarity measure from which redundant operations have been eliminated.

Figure 5 is a flow chart for determining the MPC similarity measure in accordance with an embodiment of the present invention.

Figure 6 shows an apparatus for determining a disparity map  $D(x,y)$  in accordance with an embodiment of the present invention.

Figure 7A illustrates the internal structure of the P-buffer of the apparatus shown in Figure 6.

Figure 7B illustrates the internal structure of the V-buffer of the apparatus shown in Figure 6.

Figure 8A illustrates the internal structure of the P-unit of the apparatus shown in Figure 6

Figure 8B is a circuit diagram of the D\_P unit of the P-unit illustrated in Figure 8A.

Figure 9 illustrates the internal structure of one of the MPC-units of the apparatus shown in Figure 6.

Figure 10 is a logic diagram of the MPC-unit whose internal structure is illustrated in Figure 9.

Figure 11 illustrates the internal structure of the Max\_MPC selector of the apparatus shown in Figure 6.

#### Best Mode for Carrying out the Invention:

In the method described and claimed in the present application, a matching pixel count ("MPC") is used as a measure of the similarity between a reference window in a reference image and each of a plurality of search windows in a search image. The contribution of a given matching pixel in

the search image to the similarity measure is taken to be the same irrespective of the intensity at the given matching pixel in order to ensure an accurate result near the boundary of the image. MPC is computed by counting the number of pixels in a window in one image whose corresponding pixels in another image have similar intensity values.

When the reference and search windows satisfy the epipolar constraint,  $MPC(x, y, d)$  denotes the similarity between a reference window centered at  $R(x, y)$  and a search window centered at  $L(x+d, y)$ , where  $R(x, y)$  is a pixel in the reference image located at  $(x, y)$  and  $L(x+d, y)$  is a pixel in the search image located at  $(x+d, y)$ .  $MPC(x, y, d)$  is defined by:

$$MPC(x, y, d) = \sum_w P(x, y, d) \quad (1)$$

where  $P(x, y, d) = 1$ , if  $\text{abs}[B_R(x, y) - B_L(x+d, y)] \leq Th$   
 $= 0$ , otherwise;

$\text{abs}(\cdot)$  denotes the absolute value;  $B_R(x, y)$  and  $B_L(x+d, y)$  denote the intensity at  $R(x, y)$  and  $L(x+d, y)$ , respectively;  $W$  denotes a matching window of size  $(W_x * W_y)$ , a group of pixels centered at either  $R(x, y)$  in the reference image or  $L(x+d, y)$  in the search image; and  $Th$  is a predetermined threshold value.

$P(x, y, d)$  is thus set equal to one if intensity values at corresponding pixels, i.e.,  $R(x, y)$  and  $L(x+d, y)$ , in the two images are similar, and zero otherwise. To determine the disparity for a reference pixel  $R(x, y)$ ,  $MPC(x, y, d)$  is computed for all values of  $d$  in a search range ( $d=0$  to  $S_r$ ) and the disparity is defined to be the value of  $d$  that yields the greatest value of  $MPC(x, y, d)$ .

Figure 2 illustrates a comparison of SAD and MPC similarity measures. A comparison of the intensity values of

pixels in the reference image with the intensity values of pixels in the search image shows that the disparity of the three upper pixels in the reference image, which have intensity values of 10, 200, and 10, is zero and the disparity of the six lower pixels in the reference image, which have intensity values of 150, 80, 60, 70, 200 and 250, is one.

To determine the disparity of the pixel with an intensity value of 80 (the "reference pixel"), a similarity measure (either SAD or MPC) between a window in the reference image (the "reference window") and each window in the search image ("search window") is computed and the search window most similar to the reference window is selected. The displacement between the reference window and the selected search window is the disparity between the two images for the reference pixel. For sake of simplicity, only two search windows are shown in Figure 2.

In the SAD method,  $SAD(3,2,0)$  and  $SAD(3,2,1)$  are given by:

$$\begin{aligned} SAD(3,2,0) &= \text{abs}(10-10) + \text{abs}(200-200) + \text{abs}(10-10) \\ &\quad + \text{abs}(140-150) + \text{abs}(150-80) + \text{abs}(80-60) \\ &\quad + \text{abs}(80-70) + \text{abs}(70-200) + \text{abs}(200-250) \\ &= 290 \text{ and} \end{aligned}$$

$$\begin{aligned} SAD(3,2,1) &= \text{abs}(200-10) + \text{abs}(10-200) + \text{abs}(250-10) \\ &\quad + \text{abs}(150-150) + \text{abs}(80-80) + \text{abs}(60-60) \\ &\quad + \text{abs}(70-70) + \text{abs}(200-200) + \text{abs}(250-250) \\ &= 620. \end{aligned}$$

In  $SAD(3,2,0)$  and  $SAD(3,2,1)$ , 3 and 2 are the x and y coordinates, respectively, of the reference pixel and 0 or 1 is the disparity between reference and search windows for which the SAD is computed. The disparity for the reference pixel in the SAD method is the displacement that yields the smallest SAD value (i.e., 0).

In the MPC method,  $MPC(3,2,0)$  and  $MPC(3,2,1)$  are given by:

$$MPC(3,2,0) = 1+1+1+0+0+0+0+0+0=3$$

$$MPC(3,2,1) = 0+0+0+1+1+1+1+1+1=6$$

The disparity for the reference pixel in the MPC method is the displacement that yields the largest MPC value (i.e., 1).

In this example, the SAD measure yields a misleading disparity value, because the upper three pixels in the reference window, which exhibit a large variance in their intensity values, have a greater effect on the similarity measure than do the lower six pixels, which exhibit a small variance.

Figures 3A and 3B illustrate redundant operations which are inherent in the computation of the MPC measure. Despite the epipolar constraint, the amount of computation required to determine MPC values for all pixels in the reference image is proportional to  $(I_x * I_y) * (W_x * W_y) * S_r$ , where  $(I_x * I_y)$  is the size of an image,  $(W_x * W_y)$  is the size of a window,  $W_x$  and  $W_y$  being odd numbers, and  $S_r$  is a search range. The enormous amount of computation required is due to the redundant operations which are inherent in the MPC method.

As shown in Figure 3A, after the MPC value between  $R(x, y)$  and  $L(x+d, y)$  has been determined, the computation corresponding to the overlapping area of the two horizontally displaced windows would be redundant when determining the MPC value between  $R(x+1, y)$  and  $L(x+1+d, y)$ . Similarly, when determining the MPC value between  $R(x, y+1)$  and  $L(x+d, y+1)$ , the computation corresponding to the overlapping area of two vertically displaced windows in Figure 3B would be redundant.

By using buffers to store the results of computation, these redundant operations can be avoided so that the amount of computation can be reduced to  $(I_x * I_y) * S_r$ , irrespective of the size of a window.

Figures 4A and 4B illustrate a method of computing MPC values from which these redundant computations have been eliminated. For sake of simplicity, only pixels in the



reference image are shown.  $V(x, y, d)$  represents a matching pixel count between two vertical segments which are a group of  $W_y$  pixels centered at  $R(x, y)$  and  $L(x+d, y)$ .  $MPC(w_x, y, d)$  is computed by summing  $V$  values as follows:

$$MPC(x_w, y, d) = \sum_{i=-w_y}^{w_y} V(w_x+i, y, d) \quad (2)$$

For  $x$  larger than  $w_x$ ,  $MPC(x, y, d)$  can be computed from a previously computed  $MPC$  value (i.e.,  $MPC(x-1, y, d)$ ) as follows:

$$MPC(x, y, d) = MPC(x-1, y, d) + V(x+w_x, y, d) - V(x-1-w_x, y, d) \quad (3)$$

In Equations (2) and (3),  $w_x$  and  $w_y$  denote distances from the center to the boundary of a window in the horizontal and vertical directions, respectively:

$$w_x = (W_x - 1) / 2 \text{ and } w_y = (W_y - 1) / 2 \quad (4)$$

In a similar manner, a first  $V$  value in a column,  $V(x, w_y, d)$ , is computed by summing  $P(\cdot)$  for a vertical segment:

$$V(x, w_y, d) = \sum_{i=-w_y}^{w_y} p(x, w_y+i, d) \quad (5)$$

where  $P(x, y, d)$  has the value one if values of  $R(x, y)$  and  $L(x+d, y)$  are similar and has the value zero otherwise as defined in Equation. (1). For values of  $y$  greater than  $w_y$ ,  $V(x, y, d)$  can be computed by using a previously computed  $V$  value,  $V(x, y-1, d)$ :

$$V(x, y, d) = V(x, y-1, d) + P(x, y+wy, d) - P(x, y-1-wy, d) \quad (6)$$

At the beginning of stereo disparity estimation for a reference image and a search image,  $V(x, wy, d)$ , which represents the similarity between vertical segments centered at  $R(x, wy)$  and  $L(x+d, wy)$ , is computed for all values of  $x$  and  $d$  (i.e., from  $x=0$  to  $x=Ix-1$  and from  $d=0$  to  $d=Sr$ ) according to Equation. (5). In other words, as many as  $(Sr+1)$   $V$  values,  $V(x, wy, 0)$ ,  $V(x, wy, 1)$ , ...,  $V(x, wy, Sr)$ , are computed for each  $x$ . The computed  $V$  values are stored in a  $V$  buffer which can store  $Ix*(Sr+1)$   $V$  values.

In Equation (2),  $MPC(wx, y, d)$  represents the similarity between a window centered at  $R(wx, y)$  in the reference image and a window centered at  $L(wx+d, y)$  in the search image. A buffer for storing a  $MPC$  value is initialized with a sum of  $V(wx+i, y, d)$  values for  $I=-wx$  to  $wx$ . The initialization of the  $MPC$  buffer is done for a  $(wx)$ th pixel in each row.

For values of  $x$  greater than  $wx$ ,  $MPC(x, y, d)$  is computed by using  $MPC(x-1, y, d)$ , the  $MPC$  for the previous pixel in the same row, which is stored in the  $MPC$  buffer, as shown in Equation (3) and Figure 4A. Specifically,  $V(x+wx, y, d)$ , a  $V$  value for a vertical segment newly added to the window, is added to the previously computed  $MPC$  value and  $V(x-wx-1, y, d)$ , a  $V$  value for a vertical segment excluded from the window, is subtracted from the previously computed  $MPC$  value.

Similarly, for values of  $y$  greater than  $wy$ ,  $V(x, y, d)$  can be computed without redundancies in the vertical direction.  $P(x, y+wy, d)$  indicating matching between pixels added to the vertical segment is added to the previously computed  $V$  value,  $V(x, y-1, d)$ , stored in the  $V$  buffer, and  $P(x, y-1-wy, d)$  indicating matching between pixels excluded from the vertical segment is subtracted from  $V(x, y-1, d)$ .

In summary, after initializing the V buffers with the V values for vertical segments centered at each pixel in the (wy)th row and initializing the MPC buffer with the MPC value for a (wx)th pixel in a row, other MPC values may be determined without redundant operations, as indicated by Equations (3) and (6). In doing this, search range for each pixel should be identical. For a pixel in the reference image  $R(x,y)$ , if a best matching pixel among all the pixels of the search image in the search range is  $L(x+d_{\max}, y)$ ,  $d_{\max}$  is the stereo disparity for  $R(x,y)$ .

Figure 5 illustrates the MPC matching algorithm without redundancies for implementing the above equations in an efficient manner. Inputs to this algorithm are: I) the intensity values of pixels  $R(x,y)$  and  $L(x,y)$  in the reference and search images, respectively, where x ranges from 0 to  $I_x-1$  and y ranges from 0 to  $I_y-1$ ; ii)  $W_x$  and  $W_y$  representing the size of a matching window,  $W_x$ ,  $W_y$  being odd integers; and iii) the search range  $S_r$ . An output from this algorithm is a disparity map containing a disparity for each pixel,  $DM(x,y)$ , where x ranges from  $w_x$  to  $I_x-w_x$  and y ranges from  $w_y$  to  $I_y-w_y$ .

In Step S20, y is initialized to  $w_y$ . Then, in Step S30, initial V values,  $V(x, w_y, d)$  for  $x=0$  to  $I_x-1$ ,  $d=0$  to  $S_r$ , are computed and the results are stored in the V buffers. Since the x coordinate of a pixel cannot exceed  $I_x-1$  and  $R(x+d, y)$  is used in the computation, x ranges from 0 to  $I_x-1-d$ , instead of  $I_x-1$ .

In Step S100, the MPC values are either initialized or updated, depending on the value of x, and the V values are updated. This step is executed for each value of x, y and d ( $x=w_x$  to  $I_x-w_x$ ,  $y=w_y$  to  $I_y-w_y-1$ ,  $d=0$  to  $S_r$ ). If  $x=w_x$ , Equation (3) is executed to generate the first MPC value in the row. If the value of x is less than  $I_x-d-w_x$  and greater than  $w_x$ , Equations (3) and (6) are executed. If x is greater than or

equal to  $I_x-d-w_x$ , only the V values are updated. After y is incremented, the V values updated in S100 are used to compute the MPC values for the next row.

5 Steps S40, S50, S60, S70, S80, S90, and S95 repeat Step S100 for the given ranges of x, y and d.

10 Steps S130 and S140 determine the value of d that yields the greatest MPC value as the disparity for the reference pixel. After all the steps have been completed, a depth map  $DM(x,y)$  (for  $x= w_x$  to  $I_x-w_x$ ,  $y=w_y$  to  $I_y-w_y$ ) is obtained.

15 Figure 6 illustrates an apparatus for determining the disparity values in accordance with the present invention. The apparatus includes a P-buffer 100, an MPC processor 200 and a V-buffer 300 in which  $W_x$  and  $W_y$  are set to seven, so that the size of a window is  $7 \times 7$ , and  $S_r$  is set to 63. Inputs to the apparatus are pixel values of the reference and search images and Th for determining whether two pixels are similar. Synchronization signals, HSync and VSync are also input to the apparatus. Outputs from this apparatus are the maximum MPC(x,y,d) value (MaxV) for pixel R(x,y) in the reference image and disparity  $DM(x,y)$  which yields the maximum MPC value.

25 The MPC processor 200 includes a P-unit 210, MPC-units 220, a Max\_MPC selector 230 and a control unit 240. The MPC processor 200 includes  $(S_r+1)$  MPC units 220, so that for each reference pixel R(x, y),  $(S_r+1)$  MPC values for a search range are computed in parallel. Pixel values R(x,y) and L(x+d,y) are fed to the P-unit 210, where P(x,y,d) is determined according to Equation (1). In the P-unit 210,  $(S_r+1)$  P(x,y,d)-values for a given (x,y) ( $d=0$  to  $S_r$ ) are determined in parallel. The P-values are provided to the P-buffer 100 and stored therein.

35 The P values stored in the P-buffer 100 are coupled to the  $(S_r+1)$  MPC-units 220, where MPC(x,y,d) and V(x,y,d) are computed. A vector  $P_v(d)$ , a set of P-values for a same d

value, are input to the MPC-units 220 from the P-buffer 100

$V(x,y,d)$  computed at the MPC units 220 ( $V_0$  in the figure) is input to the V-buffer 300 and stored therein. Each of the MPC-units 220 computes the MPC value for each  $d$  by using a  $V$  value computed therein and  $V$  values retrieved from the V-buffer 300. To do this,  $V_a$  and  $V_i$ ,  $V$  values for vertical segments added to and excluded from a matching window, respectively, are provided from the V-buffer 300 to each of the MPC-units 220.

Max\_MPC selector 230 receives  $(S_r+1)$  MPC values for a pixel  $R(x,y)$  from the MPC-units 220 and determines a largest one ( $MaxV$ ) to provide  $DM(x, y)$ .

Figure 7A illustrates the internal structure of the P-buffer 100, which includes  $(S_r+1)$  parallel  $(I_x \times W_y)$  bit shift registers. The P-buffer stores the  $P(x,y,d)$  values (identified as  $P_i$  in the figure) provided from the P-unit 210 in the MPC processor 200 and timely provides them to the  $(S_r+1)$  MPC-units 220. Each of the blocks identified as  $P\_R(x,y)$  stores 64 (i.e.,  $(S_r+1)$ ) 1-bit  $P(x,y,d)$ -values and the  $P(x,y,d)$ -values are shifted in synchronization with a system clock, as indicated by the arrows. In Figure 7A, each row of blocks corresponds to a row of an image and each column of blocks corresponds to a vertical segment in a window (i.e.,  $W_y$  consecutive pixels in a column). Thus,  $P(x,0,d)$  to  $P(x,W_y-1,d)$  P-values for pixels included in the same vertical segment are provided from the left part of the P-buffer 100, where each of  $P(x,0,d)$  to  $P(x,W_y-1,d)$  represents 64 P-values (either 0 or 1) for  $d=0$  to 63. These values are realigned according to the  $d$  values into  $S_r+1$   $P_v(d)$  so that  $W_y$  (in this example, 7) P values for a same  $d$  constitute a vector  $P_v(d)$ . Since the P-buffer 100 is constructed with shift registers, desired results are provided when the registers are fully occupied by P values, that is,  $I_x \times W_y$  clocks after a data is first fed to the P-buffer. Then,  $P_v(d)$ s are provided from the P-buffer 100 consecutively. Each  $P_v(d)$  from the P-buffer 100 is fed back to each MPC-unit in the MPC-processor 200 to be used in

computing V.

Figure 7B shows the internal structure of the V-buffer 300. The V-buffer 300 temporarily stores  $V(x,y,d)$  values which are computed at the  $(S_r+1)$  MPC-units 220. Each block identified as  $V\_R(I)$  stores 64 ( $d=0$  to 63) 3-bit  $V(x,y,d)$  values. The data in each block is shifted to a next block in synchronization with a clock signal. Data Stored in  $V\_R(W_x)$  and  $V\_R(0)$  are provided as outputs from the V-buffer 300, which correspond to V values for vertical segments added to and subtracted from a matching window, respectively. In other words, each of  $V_w(0)$  to  $V_w(63)$  represents a V value for a vertical segment added to a window for  $d=0$  to 63 in the computation of MPC. Similarly, each of  $V_o(0)$  to  $V_o(63)$  represents a V value for a vertical segment excluded from a window for  $d=0$  to 63. The V-buffer 300 also has initial delay of  $(I_x-W_x)$  clock cycles. Outputs from the V-buffer 300 are provided to the MPC-unit 220 to be used in computing MPC values according to Equation (3).

Figure 8A shows an internal structure of the P-unit 210 included in the MPC-processor 200. Inputs to the P-unit 210 are R, S and Th where R and S denote pixel values of reference and search images, and Th denotes the threshold for determining  $P(x,y,d)$ . Assuming that a pixel value ranges from 0 to 255 ( $2^8-1$ ), 8 bits are assigned to each R and S input. The P-unit 210 receives  $(S_r+1)$  pixel values in a search image for each pixel in a reference image and determines  $P(x,y,d)$  for them. A processing unit denoted as  $D\_P(d)$ ,  $d=0$  to  $S_r$ , determines  $P(x,y,d)$  for the reference pixel (i.e., the pixel in the reference image) and a pixel in the search image displaced from the reference pixel by d.

Figure 8B is a circuit diagram of the  $D\_P$  unit 211. Full adders (FA) 211-1 in the left part subtract S from R. Carry generators (CG) 211-3 in the right part provide a carry, denoted as D, depending upon the sign of  $(Th - \text{abs}(R-S))$ . If Th

is greater than the absolute value of  $(R-S)$ ,  $D$  is 1. If  $(R-S)$ , the input to the XOR gates 211-2, is a positive number, the 2's complement of the inputs (i.e.,  $-(R-S)$ ) is provided by the gates 211-2. If  $(R-S)$  is a negative number,  $(R-S)$  is provided by the XOR gates, so that the absolute value of  $(R-S)$  can be subtracted from  $Th$ . Since there are  $(Sr+1)$   $D\_P(d)$  units 211 in the  $P\_units$  210,  $(Sr+1)$   $P(x,y,d)$  values may be calculated in parallel. In the  $P$ -unit 210 shown in Figure 8A, a pixel value in a reference image is used only once, while a pixel value in the search image is used  $(Sr+1)$  times as it is shifted through  $D\_R(d)$ .  $D\_R(d)$  denotes the registers that store pixel values of the search image. A pixel value in each  $D\_R(d)$  is shifted left on each clock. As the  $P$ -unit 210 includes the shift registers, it provides an output continuously after an initial delay of  $(Sr+1)$  clock cycles.

Figure 9 illustrates the internal structure of the MPC-unit 220 in the MPC processor 200. The MPC-unit 220 is divided by its function into a  $V\_MP$  counter 221 and a  $V\_MP$  update unit 222 for calculating  $V(x,y,d)$  according to Eqs. (5) and (6), respectively, and a  $W\_MP$  count and update unit 223 for calculating  $MPC(x,y,d)$  according to Eqs. (2) and (3). The  $V\_MP$  counter 211 is used to provide  $V(x,y,d)$  by summing  $P$  values inputted in an initial stage of the disparity map computation.

After the initial stage,  $V$  (denoted as  $V_i$  in the figure) is provided to the  $V\_MP$  update unit 222 from the  $V$ -buffer 300. Then,  $V\_MP$  update unit 222 calculates  $V(x,y,d)$  without redundant operation.

Figure 10 depicts a logic diagram of the MPC-unit 220. The  $V\_MP$  counter 221 receives  $W_y$   $P$  values in an initial stage, adds them at four full adders (FAs) included in the  $V\_MP$  counter 221, thereby providing a 3-bit binary number  $V(x,y,d)$ . This 3-bit value ( $Vo(2)$ ,  $Vo(1)$ ,  $Vo(0)$ ) is provided to the  $V$ -buffer 300 through a MUX A 225.  $C2$  control input makes the MUX A 225 select an output from the  $V\_MP$  counter 221 until an output from the  $V\_MP$  update unit 222 becomes valid.  ~~$V(x,y,d)$~~

values input to the V-buffer 300 are provided to the Va input port of the MPC-unit 220 after (Ix-Wx) clock cycles and to the Vi input port after Ix clock cycles. V\_MP update unit 222 begins to calculate V(x,y,d) without redundancy according to Equation (6) by adding one to Vi or subtracting one from Vi at the full adders 222-3, depending on the values of Pv(0) and Pv(6). Specifically, if Pv(0) is equal to Pv(6), 0 (000 in binary) is added to Vi; Pv(0) is 1 and Pv(6) is 0, -1 (111 in binary) is added to Vi; and Pv(0) is 0 and Pv(6) is 1, 1 (001 in binary) is added to Vi. Then, the C2 control input enables the MUX A 225 to select the outputs from the V\_MP update unit 222.

Vi and Va, provided from the V-buffer 300 to the MPC-unit 220, are used to calculate MPC values according to Equations (2) and (3). Full adders 223-1 on the left part of the W\_MP count and update unit 223 calculate Va-Vi. The result is added to a previous MPC value in the adders 223-2 on the right part of the W\_MP count and update unit 223 in order to provide an updated MPC value. MUX B 224 selects 0's and provides them to the W\_MP count and update unit 223 until Vi from the V-buffer 300 becomes valid. As a result, until Vi becomes valid, only Va values are summed in the unit 223 to provide MPC(wx, y, d) according to Equation (2). After Vi from the V-buffer becomes valid, MPC is calculated according to Equation (3).

Figure 11 shows the internal structure of the Max\_MPC selector 230 included in the MPC-processor 200 in Figure 6. The Max\_MPC selector 230 compares MPC values provided from the (Sr+1) MPC-units 220, selects a maximum value MaxV and provides a displacement DM(x,y) corresponding to the maximum value. Basic units for constructing the Max\_MPC selector 230 are C&A(n) cells 231 each of which compares its inputs and provides the larger one and its location information.

Figure 12 is a logic diagram of the C&A(n) cell. In the C&A cell 231, carry generators 231-1 provides 0 if Ib is



smaller than Ia and 1 otherwise so that MUXes 231-3 select and provide the larger value. MUXes 231-2 produce the location information ( $S(n)$  , ... ,  $S(0)$ ) corresponding to the larger value in response to the location information from a previous stage, i.e.,  $C\&A(n-1)$ , and the carry value. The number of MUXes in the C&A cell varies depending on  $n$ , that is, the number of MUXes 231-2 is  $n$ . The MUXes 231-2 select one between  $S_a$  and  $S_b$ , the location information of a previous stage. In Fig. 11, going further to the right, number of bits inputted as  $S_a$  and  $S_b$  to C&A cell increases. The output from the C&A cell of the last stage is a maximum MPC value for a given pixel  $R(x,y)$  and its disparity.

While the present invention has been described in terms of particular embodiments, it will be apparent to those of ordinary skill in the art that various changes and modifications may be made without departing from the spirit and scope of the invention defined in the following claims.